AN1269
APPLICATION NOTE

Connecting the MPC555 Microcontroller to the M58BF008 Series Flash Memory

CONTENTS

INTRODUCTION

This application note describes a method of to connecting the M58BF008 Flash memory to the MPC555 microcontroller. The M58BF008 is an advanced 8 Mbit Flash memory from STMicroelectronics, which can be configured as 256 Kbit x32 bus width. The memory includes a burst interface for high speed access, allowing code to be executed directly from the memory without the need to shadow the code in RAM.

The MPC555 is a member of Motorola's PowerPC family of integrated microprocessors. It is a general purpose 32-bit microcontroller with a wide variety of application areas but it particularly targeted towards automotive applications.

ADVANTAGES OF FLASH

Flash memories can be used to store both code and data for the MPC555 microcontroller. Unlike EPROMs the data in Flash memories can be changed by the microcontroller. This enables non-volatile user data to be stored in the Flash. Field upgrades of the application code can be performed without any disassembly, unlike EPROM solutions.

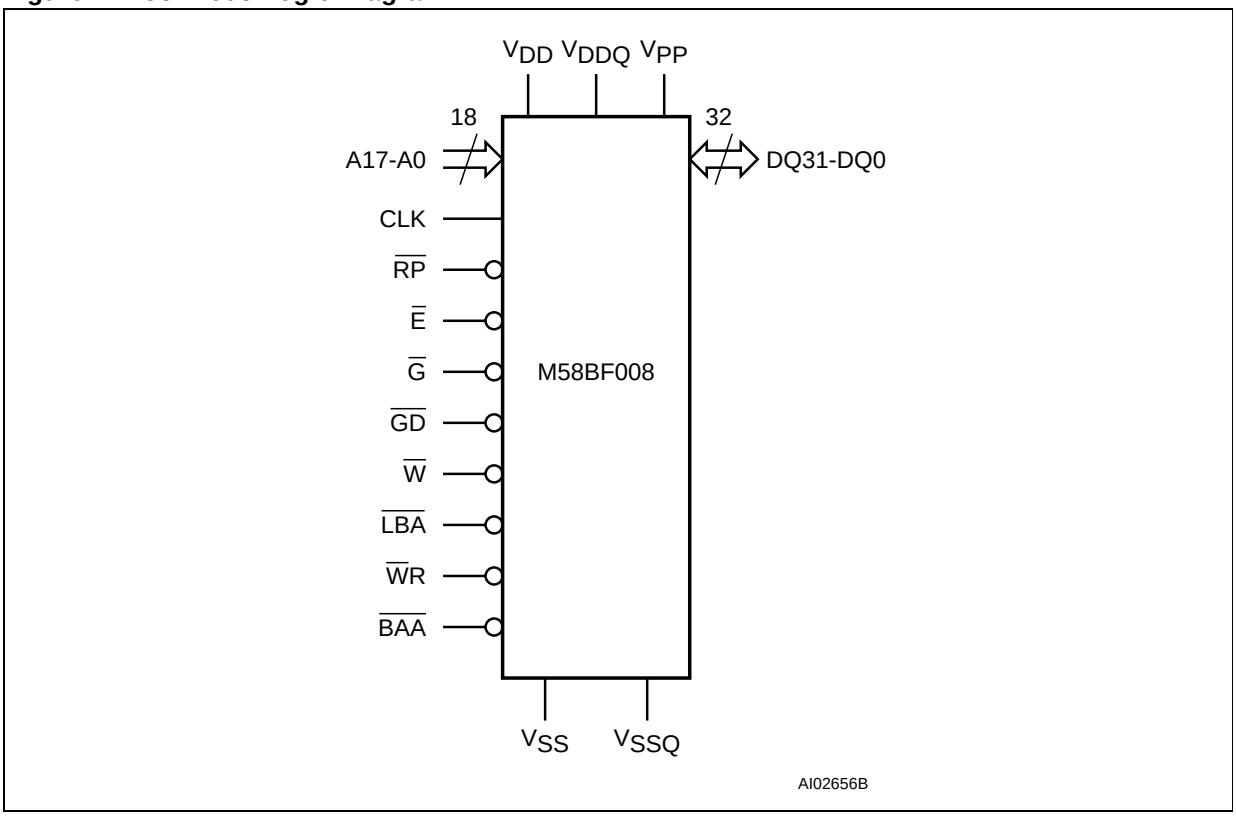
It is usual to write separate boot and application programs so that the application program can be upgraded without changing the boot program. If the upgrade fails then the processor will still boot and it will be possible to reattempt to upgrade the application. The boot code can be programmed into the Flash before the Flash is fitted to the circuit board or Debug Port can be used to boot the microcontroller and then program the Flash.

The M58BF008 Flash memory has a special block, called the overlay block, that can be used to store the boot program. This is an extra 32 Kbyte block that is mapped in place of block 0 on reset. The MPC555 can boot from this block, then disable it, giving access to block 0. The memory is 8 Mbit plus 256 Kbit in size.

M58BF008 BUS ARCHITECTURE

Consider the bus on the M58BF008, Figure 1 shows the Logic Diagram. The memory has separate Address and Data Buses that can connect directly to the Address and Data Buses on the MPC555. There are several control lines required for burst access, these are Chip Enable ($\overline{E}$), Output Enable ($\overline{G}$), Write Enable ($\overline{W}$), System Clock (CLK), Load Burst Address ($\overline{LBA}$), Write/Read ($\overline{WR}$) Burst Address Advance ($\overline{BAA}$) and Output Disable ($\overline{GD}$). Also, Reset/Power-down ($\overline{RP}$) is present.

Figure 1. M58BF008 Logic Diagram



A burst read cycle is initiated on the rising edge of the System Clock when Chip Enable is Low to select the memory, a valid address is on the Address Inputs, Latch Burst Address is Low to latch the address and Write/Read is High to indicate a read. Chip Enable and Write/Read should remain constant throughout the read operation, whereas Load Burst Address should only be low during the cycle where the address is latched, after that the address can change. Output Enable must be Low to read the data when it becomes valid (after the X-latency) and Burst Address Advance must be Low to increase the address and output new data after the Y-latency expires. The Memory Controller in the MPC555 can be configured to generate these signals without glue logic.

Unlike most RAM options, it is not possible to write a single byte or a single 16-bit word to the memory. Only 32-bit writes are supported. (This applies equally to reads, but, for a read, the unwanted data can be discarded). Commands are written on DQ0-DQ7 to the memory, with DQ8-DQ31 being ignored; the double-words are written to the memory array as part of the program operation.

The Reset/Power-down pin ($\overline{RP}$) should be held High for normal operation of the memory, this is the memory's non-reset state. To ensure that the overlay block is always read during the boot process and the memory is always in its read mode the Reset/Power-down should be Low when the MPC555 is reset. The memory can be placed in a low-power mode by holding Reset/Power-down Low; after Reset/Power-down is brought High again the memory returns to its reset state and may require configuration before being accessed correctly.

In Asynchronous modes several pins are not used and their voltage is "don't care". These are CLK, $\overline{LBA}$, $\overline{WR}$ and $\overline{BAA}$. During asynchronous accesses the CPU does not need to control these pins, they can be tied to V_{IL} or V_{IH}, or change through the access. This may be useful when trying to replace another memory with an M58BF008.

MPC555 BUS ARCHITECTURE

The MPC555's bus architecture can be daunting on first appearance. There are many control lines to allow for 8-bit accesses, 16-bit accesses, 32-bit accesses, bus arbitration, etc. Many applications do not need to make use of these features. Only a simple connection is considered here.

The MPC555 can be configured as Big-Endian or Little-Endian, with the normal configuration being Big-Endian. The M58BF008 part is Little-Endian insofar as the hexadecimal data used to control the command interface uses D0 to equate to the LSB. To use the M58BF008 in the Big-Endian mode of the MPC555 the Address bus and the Data bus should be reversed; A29 on the MPC555 should be connected to A0 of the M58BF008, A28 to A1, A17 to A2, etc.; D31 on the MPC555 should be connected to D0 on the M58BF008, D30 to D1, D29 to D2, etc. If this convention is followed then the standard C code drivers for the M58BF008 will not have to be changed. If this convention is not followed then the data for the command interface will need translating for the before the memory's Command Interface will respond correctly and continuous data in the processor's address space will cross through each block of the M58BF008 in turn.

The MPC555's memory controller includes control lines that are suitable for connecting to the M58BF008 in either synchronous or asynchronous mode, or both, without the need for glue logic. Each transfer is initiated with the address being placed on the address bus, Transfer Start ($\overline{TS}$) Low and the Chip Select ($\overline{CSx}$) Low during the rising edge of the System Clock. Output Enable ($\overline{OE}$) or Write Enable/Byte Enable ($\overline{WE}$/$\overline{BE}$0) falls Low to indicate a read or a write. In burst read operations Burst Data In Progress ($\overline{BDIP}$) indicates that the memory should advance the address and Read/Write (RD/$\overline{WR}$) gives advance notice that the burst operation is a read or a write (the MPC555 never uses burst write operations, but other bus masters are allowed to).

Burst operations on the MPC555 read from sequential (not interleaved) addresses. For example, if a 32-bit wide burst operation starts from address 1234h, the next address read will be 1238h, followed by 123Ch. The MPC555 does not issue continuous or wrapped burst operations, so in the example the fourth burst (to address 1240h) would only occur as part a separate operation. Up to four 32-bit reads are possible in one burst operation, but only when the four LSBs of the address start at 0000b (i.e. from address 1230h in the example).

Although the MPC555 is a 3.3V processor, its external bus is 5V tolerant. 5V parts can be connected to the external bus without the need for transceivers. The short rise and fall times mean that the timings when connecting to 5V parts are not significantly different, though some extra consideration may be required for rising signals. Table 1 gives a summary of the Input/Output DC characteristics on the high and low voltage levels and shows that they are compatible.

Table 1. Input/Output DC Characteristics Comparison

Comment	MPC555		M58BF008	
MPC555 Low Input from M58BF008	V_{IL3M}	0.8V max	V_{OL}	0.2V max
MPC555 High Input from M58BF008	V_{IH3M}	2V min	V_{OH}	3.1V min
M58BF008 Low Input from MPC555	V_{OL3M}	0.5V max	V_{IL}	0.8V max
M58BF008 High Input from MPC555	V_{OH3M}	2.4V min	V_{IH}	2V min

Booting the MPC555 from External Flash

The boot mode of the MPC555 is selected by driving the data bus into a specific state during reset with Reset Configuration ($\overline{RSTCONF}$/TEXP) Low. To boot from an external memory (e.g. Flash) the internal Flash should be disabled (D20 Low), the external boot should be enabled (D3 Low) and the Boot Port Size should be set to the memory width (D4-5 both Low for 32-bit width).

For an external boot the MPC555 uses Chip Select 0 ($\overline{CS0}$) to select the boot memory; $\overline{CS0}$ is active (Low) for every memory address during the boot process until the registers that configure it are changed. 15 wait states are selected and asynchronous read operations are used.

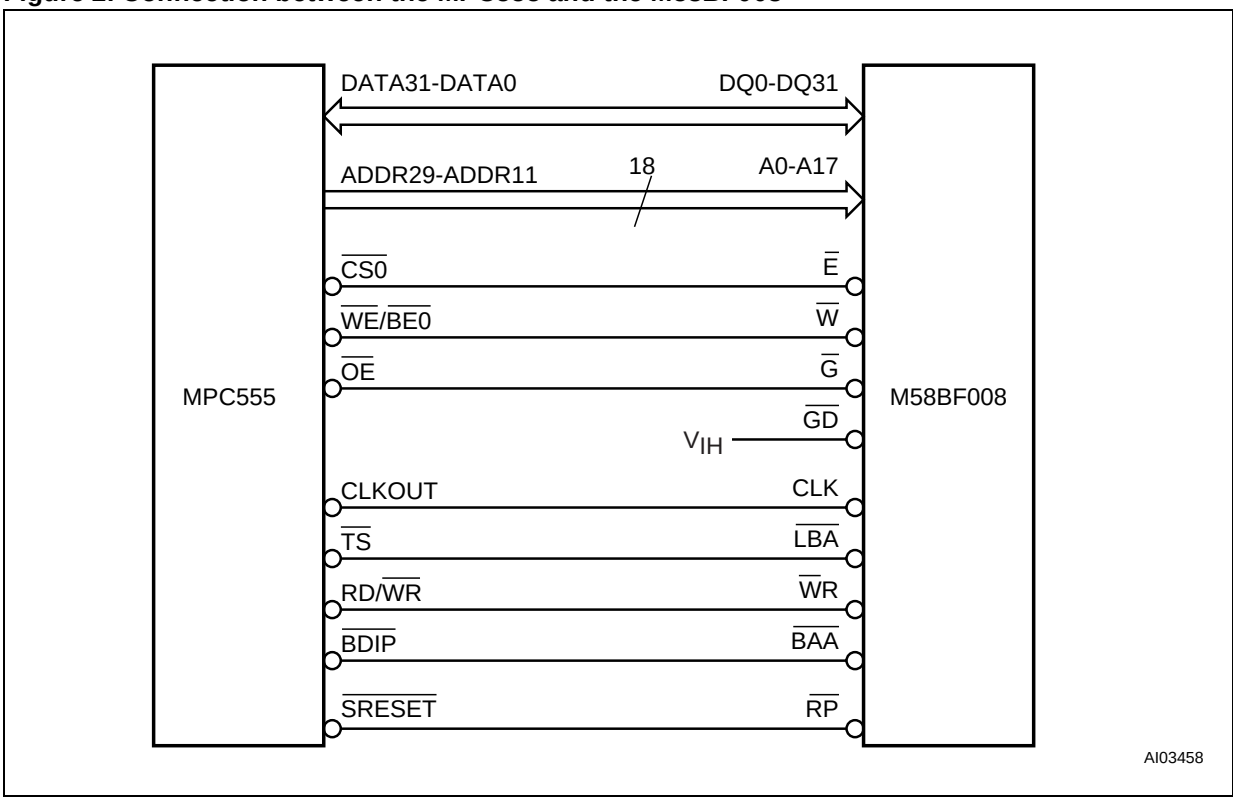
After a Hardware Reset ($\overline{HRESET}$) the boot process executes the instruction at address 00000100h. Note that this instruction is at memory location 00040h in the M58BF008's address space since it is a 32-bit part. Generally the instruction at address 00000100h will be a jump to the location in the MPC555's memory space where the Flash will be mapped. From here the boot process can continue, the Base Registers and Option Registers can be configured correctly so that other devices on the bus can be accessed.

MPC555 TO M58BF008 CONNECTION

Figure 2 shows a connection scheme for the MPC555 to M58BF008. In this configuration the M58BF008 can be the boot device. Note the reversal of the address bus and the data bus between the two devices to account for the Big-Endian/Little-Endian change. Also note that A0 of the M58BF008 should be connected to A29 of the MPC555; A30 and A31 address words and bytes, they are not used for double-word width devices. If the M58BF008 is not going to be the boot device then any of the other chip selects can be used instead. Although $\overline{GD}$ is shown connected to V_{IH} in this implementation, it can be left floating since it has an internal pull-up resistor that will ensure correct operation.

To access the memory correctly the registers in the MPC555 should be set up according to Table 4. Note that the Motorola definition for describing the X-latency and Y-latency is different to the STMicroelectronics definition. Motorola count the number of clock cycles between two successive accesses, or wait-states, whereas STMicroelectronics count the difference between the clock numbers. So, for example, a 2 wait-state access (SCY field of the Base Register set to 02h) with burst beats of 0 wait-states (BSCY field set to 00h) is equivalent to a 3.1.1.1 burst access in STMicroelectronics terms. Care should be taken when configuring the Memory Controller in the MPC555 to ensure that the correct and optimum values are used.

Figure 2. Connection between the MPC555 and the M58BF008



MPC555 Bus Configuration Registers

Three registers are used to configure the bus on the MPC555. For each chip select there are two registers, a Base Register and an Option Register; additionally, to enable burst operations the BBC Module Configuration Register must be configured correctly. The bits in each of these registers is described in detail in the MPC555 user manual; the description here is tailored specifically to using the MPC555 with an M58BF008. Table 2 gives an description of the Base Register bits and Table 3 gives a description of the Option Register bits.

Table 2. Base Register Description

Bit(s)	Mnemonic	Name	Description
0-16	BA	Base Address	The upper bits of the address are compared to the bits in this register. Bits that are masked in the Option Register do not affect the output of the comparison. If all the unmasked bits match then the corresponding chip MPC555 selected the device attached to it. Each chip select is active throughout a 64 Kbyte or more section of the memory. If the M58BF008 is using the chip select then these bits set the address where the M58BF008 is mapped in the MPC555 memory space.
17-19	AT	Address Type	These bits select whether the device on the chip select should respond to all types of access (Instruction/Data, User/Supervisor, etc.) or not. The mask in the option register controls whether the address type is ignored or not.
20-21	PS	Port Size	The port size determines whether an 8-bit, 16-bit or 32-bit device is connected to the data bus. The M58BF008 is a 32-bit memory and should use the 32-bit port setting (00b).
22			Reserved
23	WP	Write Protect	The device can be designated as read-only through this bit. The M58BF008 can be written to and, even if the user wanted to protect the memory, it is not advisable to use this method to protect it as it would disable other features, such as Read Electronic Signature. The M58BF008 should be configured as a read/write part for usual operation.
24-25			Reserved
26	WEBS	Write-Enable/ Byte Select	The output pins $\overline{WE}$n/$\overline{BE}$n can either be used to select individual bytes in a word or double-word or as a Write Enable line. In the M58BF008 it is not possible to select individual bytes, therefore the pins should be configured as Write Enable outputs.
27	TBDIP	Toggle Burst Data in Progress	Determines how long the $\overline{BDIP}$ output will be Low during each data beat of a burst cycle. The correct setting for the M58BF008 is 0.
28	LBDIP	Late Burst Data in Progress	The $\overline{BDIP}$ output can be configured to go low at the start of the burst transfer or after the X-Latency has expired. Some devices require $\overline{BDIP}$ late in order to trigger the output of the data and update of their internal address register at the correct time. The M58BF008 however will ignore the $\overline{BAA}$ input until the preconfigured X-Latency has expired. The recommended setting is for $\overline{BDIP}$ early, though both will work. ($\overline{BDIP}$ early alleviates the need to check the timing!)
29	SETA	External Transfer Acknowledge	Some devices can set the $\overline{TA}$ on the MPC555 pin to acknowledge the end of a transfer. The M58BF008 does not generate this signal, the internally generated signal from the MPC555 memory controller will terminate the cycle at the correct time.
30	BI	Burst Inhibit	The memory controller in the MPC555 needs to be told whether the device can support burst transfers or not. The MPC555 does support burst transfers. When used in asynchronous mode this bit should be set to ‘1’ to inhibit the burst, in burst mode set this bit to ‘0’ to enable the burst.
31	V	Valid Bit	The valid bit configures the memory controller to use this chip select. The bit should be set to ‘1’.

Note: The bits here are numbered according to the Motorola MPC555 definition; Bit 0 is the MSB and bit 31 is the LSB.

Table 3. Option Register Description

Bit(s)	Mnemonic	Name	Description
0-16	AM	Address Mask	The Address Mask is used in conjunction with the Base Address in the Base Register to select the address region where the chip select is active. A bit set to '1' indicated that this address bit is used. Use these bits, in conjunction with the Base Address to map the M58BF008 in the correct memory space. The M58BF008 occupies an address space of 1 Mbyte in the memory space of the MPC555, so typically the address mask would be set to FFF0h.
17-19	AMT	Address Type Mask	The address mask allows more than one type of address space to be assigned to a single chip select. If the M58BF008 is to be used for Instructions and Data, in User and Supervisor modes and for Normal and Special transfers then all bits should be masked (set to '0').
20	CSNT	Chip Select Negation Time	The Chip Select Negation Time can be changed to ensure that the device releases the bus in time for the next device to drive it and avoid bus contention. The recommended setting for the M58BF008 is '0', normal negation time.
21-22	ACS	Address to Chip Select Setup	The chip select can be configured to be set up at the same time as the address, or after the address has been set up. To meet the timings of the M58BF008 it is essential to set the chip select up as early as possible, therefore the '00' setting should be used.
23	EHTR	Extended Hold Time on Read	It is possible to have an extra idle cycle inserted between reads and writes in difference memory controller bank. This avoid bus contention for parts that drive the bus sooner than others release it. The M58BF008 is fast enough so that it does not require an additional cycle (though other devices may require the extra cycle; their EHTR registers must be set correctly).
24-27	SCY	Cycle Length	The Cycle Length is the number of wait-states required to access the device. In burst transfers this number is determined by the X-Latency. See Table 4 for the recommended values.
28-30	BSCY	Burst Beats Length	The Burst Beats Length is the number of wait-states required to access the device between burst beats. It is determined by the Y-Latency. See Table 4 for the recommended values.
31	TRLX	Timing Relaxed	Normal or Relaxed timings can be selected using this bit. The M58BF008 does not require relaxed timings (which are slower) so a value of '0' should be used.

The recommended settings for the registers are summarized in Table 4.

Table 4. MPC555 Memory Controller Register Settings Summary

Symbol	Register	Bits	Asynchronous		Synchronous	
			2 wait-states	3 wait-states	3.1.1.1	4.1.1.1
BRx	Base Register	20-31	003h	003h	001h	001h
ORx	Option Register	20-31	020h	030h	020h	030h
BBCMCR	Burst Buffer Controller Module Configuration Register	18	X	X	1	1

Note: The SCY field of the Option Register (bits 24-27) is set to 2 for 3-1-1-1 access and 3 for 4-1-1-1 access.

The M58BF008 can operate at 4-1-1-1 with a maximum clock frequency of 40MHz and at 3-1-1-1 with a maximum clock frequency of 33MHz clock.

Other connection schemes can be used to control the M58BF008. This scheme is the recommended technique, but other connections are discussed in the examples that follow.

EXAMPLE BUS OPERATIONS

The following sections go into detail regarding the bus operations and signal interactions between the MPC555 and the M58BF008. Each section gives an example waveform and timing information. Some alternative connection details are shown for designers who are trying to use the M58BF008 as a replacement part in an circuit containing an MPC555. For new designs the recommended connections should be used.

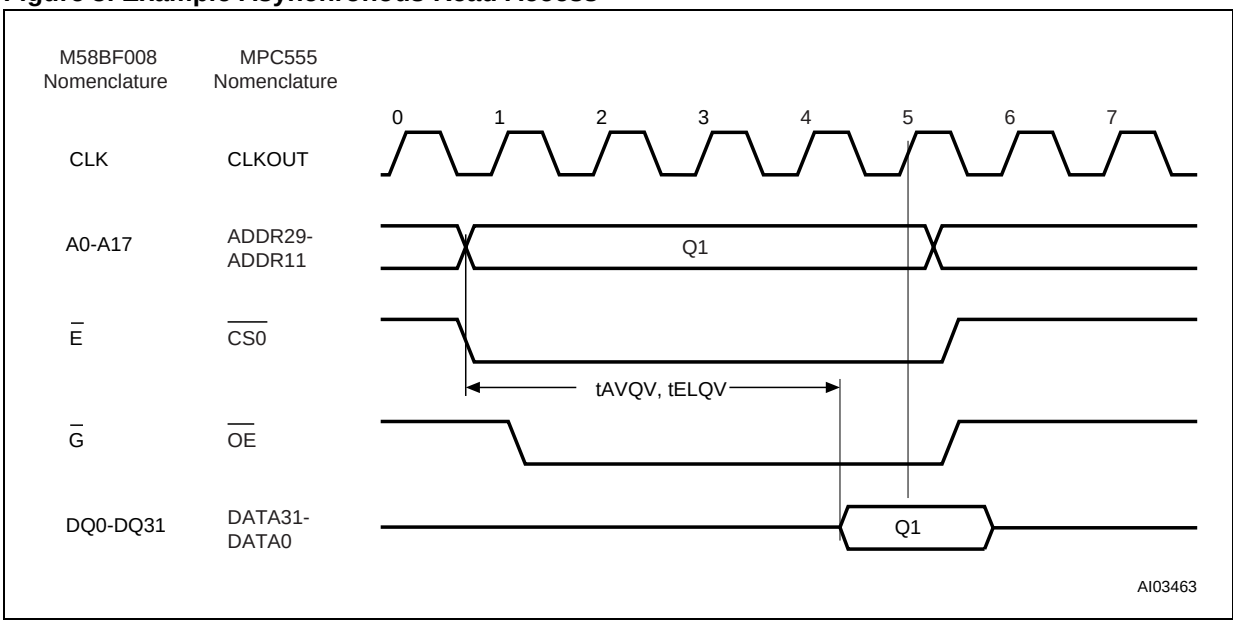
Asynchronous Burst Read Example

Figure 3 shows an asynchronous read access with three wait-states. The MPC555 sets up the address and activates the chip select ready for cycle 1 of its synchronous clock. The M58BF008 does not wait for the clock in asynchronous mode and starts fetching the data immediately. After tAVQV and tELQV (both 90ns) the data becomes valid; the MPC555 latches the data on the rising edge of clock cycle 5. Under zero wait-state operation the data is latched on clock cycle 2.

The critical timing in this example is tELQV. For a 40MHz clock (and a 40MHz MPC555) the chip select activates 14ns (timing 19) after clock cycle 0. The data must be setup 6ns (timing 17) before clock cycle 5. Using 3 wait-states clock cycle 5 is 125ns after clock cycle 0, with 14ns before the chip select and the data needing to be ready 6ns before the clock rising tELQV needs to be 105ns or less. The M58BF008 guarantees tELQV less than 90ns. The tAVQV is not required for the analysis since the address is set up 13ns (timing 8) after clock cycle 0, 1ns before the chip select; tAVQV is also 90ns for the M58BF008. For an MPC555 running at 33MHz only 2 wait-states are required.

At boot time the MPC555 uses 15 wait-states to read the boot memory. The M58BF008 boots in asynchronous mode; the 15-wait state latency is more than sufficient to successfully boot the MPC555 using an M58BF008.

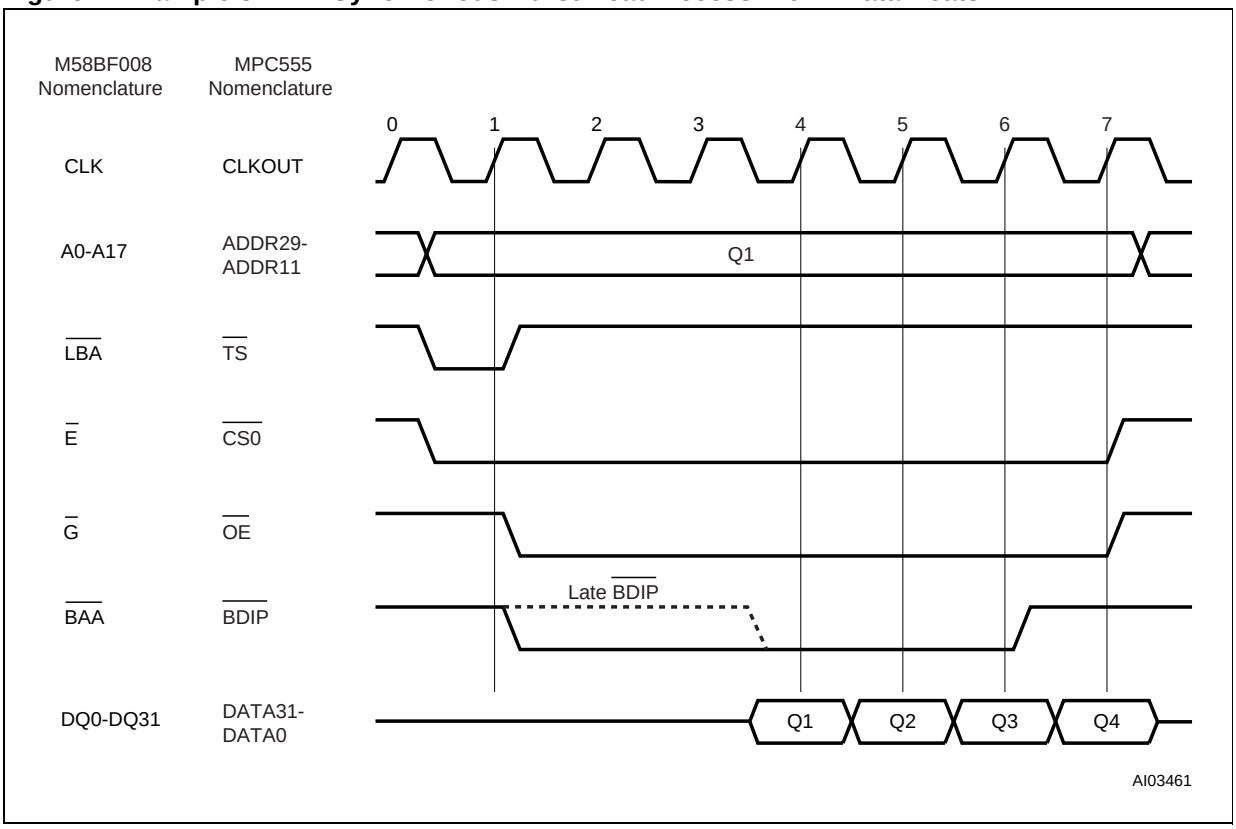
Figure 3. Example Asynchronous Read Access



Synchronous Burst Read Example

The M58BF008 toggles between asynchronous and synchronous each time the 60h command is written to the Command Interface; before the first synchronous access is performed the memory must be toggled from asynchronous to synchronous mode. Figure 4 shows the waveforms for a synchronous burst read access with four beats. The MPC555 starts by placing the address on the address bus, activating Transfer Start ($\overline{TS}$), Chip Select 0 ($\overline{CS0}$), Output Enable ($\overline{OE}$) and Burst Data In Progress ($\overline{BDIP}$). The memory uses $\overline{TS}$ to latch the address and latch the first address of the burst. After the X-latency Q1 is output on the data bus; Q2, Q3 and Q4 follow on the subsequent clock cycles. Finally Chip Select 0 ($\overline{CS0}$) rises to terminate the burst.

Figure 4. Example 3-1-1-1 Synchronous Burst Read Access with 4 Data Beats



From Figure 4 it is clear that the signals generated by the MPC555 can be used to access the M58BF008. By examining the timing requirements of the M58BF008 and comparing them to those of the MPC555 it can be shown that both the 33MHz timings and the 40MHz timings match. Table 5 shows the timings of the critical timing parameters.

Table 5. Synchronous Burst Timing Requirements

M58BF008		MPC555	
Symbol	**Timing**	**33MHz**	**40MHz**
t_{AVCH}	Min 8	15	12
t_{BLCH}	Min 8	15	12
t_{CHBH}	Min 3	7	6
t_{GLCH}	Min 10	19	17
t_{ELCH}	Min 8	14	11
t_{WRHCH}	Min 8	15	12
t_{CHQV}	18	Max 24	Max 19
t_{CHQX2}	5	Min 1	Min 1

Asynchronous Write Example

All bus operations from the MPC555 are synchronous with its clock. However, M58BF008 parts that are configured for asynchronous write operations ignore the CLK input and are accessed in an asynchronous manner. Figure 5 shows the signal waveforms for the recommended connections; three wait-states are shown. The MPC555 starts by placing the address on the address bus, activating Transfer Start ($\overline{TS}$), Chip Select 0 ($\overline{CS0}$) and Read/Write (RD/$\overline{WR}$). Write Enable ($\overline{WE}$) is asserted (Low) after the address is latched to indicate a write operation and the data to be written is placed on the data bus. After the wait-states have expired the MPC555 drives $\overline{WE0}$ and $\overline{CS0}$ High to indicate the end of the cycle before changing the address and data on the bus.

From Figure 5 it is clear that the signals generated by the MPC555 can be used to access the M58BF008. By examining the timing requirements of the M58BF008 and comparing them to those of the MPC555 the number of wait-states required can be found; Table 2 shows the timings of the critical timing parameters.

Table 6. Asynchronous Write Timing Requirements

M58BF008		MPC555	
Symbol	**Timing**	**2-wait states, 33MHz**	**3-wait states, 40MHz**
t_{AVAV}	Min 70	112	113
t_{WLWH}	Min 70	105	92
t_{DVWH}	Min 70	75	87
t_{WHWL}	Min 30	19	19

Note that the timings in Table 6 show that back-to-back write operations to the Flash are not possible, otherwise the t_{WHWL} timing is violated. It will be necessary to ensure that software does not try to perform back-to-back write operations, an extra delay (such as a non-cached NOP) may be required between writes to ensure correct operation.

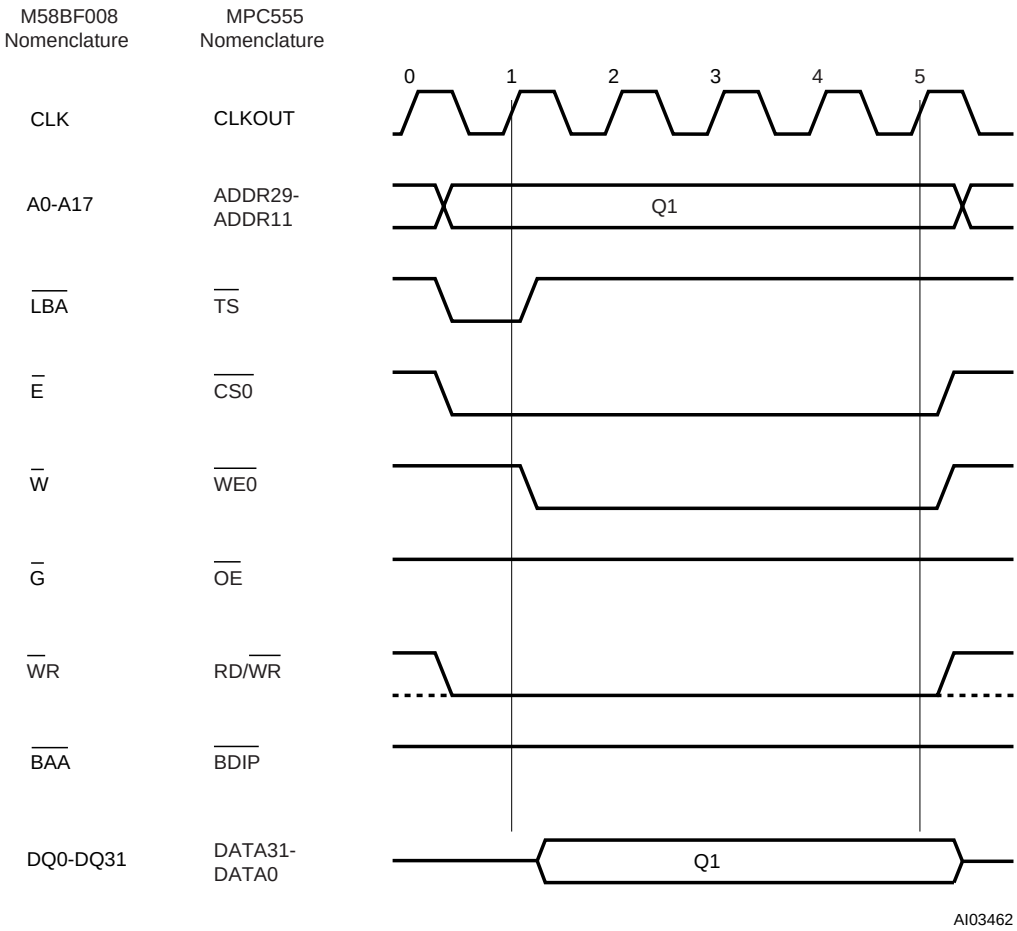
Also note that 3 wait-states are required to meet the asynchronous timing requirements at 40MHz and 2 wait-states at 33MHz.

If the M58BF008 used has been factory configured for asynchronous write operations then $\overline{W}$R, $\overline{BAA}$ and $\overline{LBA}$ are ignored during write operations. This allows various changes to the recommended connection technique to be made without loss of performance. For example:

1. The $\overline{W}$R pin on the M58BF008 can be left disconnected. A weak pull-up inside the M58BF008 will ensure that the internal level is V_{IH}, enabling the read operations to be performed correctly.
2. The $\overline{W}$R pin on the M58BF008 can be connected to $\overline{WE0}$ pin on the MPC555. Since the $\overline{W}$R input is ignored in asynchronous operations it will not affect the write operation.
3. The $\overline{W}$ pin on the M58BF008 can be connected to the $\overline{RD}$/WR pin of the MPC555, though in this case back-to-back writes are still not possible. Care should be taken because there are situations where $\overline{CS0}$ is only high for the briefest instant of time and this can be missed by the M58BF008. So long as other accesses are performed in the meantime this solution will work correctly.

Note that tying the $\overline{W}$ pin on the M58BF008 to ground will not allow the part to work correctly since a write operation will be selected whenever the part is accessed, including when read operations are intended.

Figure 5. Example Asynchronous Write Access



Synchronous Write Example

From the perspective of the MPC555 all writes are synchronous. The waveforms for synchronous write are the same as those for asynchronous writes. Therefore, the waveforms in Figure 5, Asynchronous Write Access, are equally valid for synchronous write operations.

The timing of the M58BF008 parts that have been factory configured for synchronous write operations have different timing requirements. Table 7 shows the timings of the critical parameters for synchronous write operations.

Table 7. Synchronous Write Timing Requirements

M58BF008		MPC555	
Symbol	**Timing**	**33MHz**	**40MHz**
t_{AVCH}	Min 8	15	12
t_{BLCH}	Min 8	15	12
t_{CHBH}	Min 3	7	6
t_{WRLCH}	Min 8	15	12
t_{WLCH}	Min 8	19	17
t_{CHWH}	Min 3	11	6
t_{QVCH}	Min 8	15	12
t_{CHQX}	Min 5	7	6
t_{ELCH}	Min 8	16	14

If the M58BF008 used has been factory configured for synchronous write operations then all the connections are required for correct operation. However, there are still times when some of the pins are ignored. This allows various changes to the recommended connection technique to be made without loss of performance. For example:

1. The $\overline{W}$ pin of the M58BF008 can be connected to the RD$\overline{WR}$ pin of the MPC555. This solution will correctly control the Write Enable signal of the M58BF008, so long as back-to-back writes are not performed to the M58BF008 by the MPC555.
2. The $\overline{W}$ pin may be tied to ground. The M58BF008 will work correctly because the RD$\overline{WR}$ pin will initiate the correct read or write cycle, $\overline{W}$ will be ignored. The MPC555 can still boot from the M58BF008 since RD$\overline{WR}$ is high during reset and remains high while the boot code is executed.

It is essential to have $\overline{W}$R connected, otherwise the addresses cannot be latched correctly during a write cycle (the internal pull-up would always select a read operation). $\overline{W}$R cannot be connected to $\overline{WE0}$ because $\overline{WE0}$ is too late for the addresses to be latched correctly during a write cycle.

STMICROELECTRONICS SOFTWARE DRIVERS

STMicroelectronics provide a set of drivers, written in C, manipulating all of the functions of the M58BF008. These are contained in a separate application note. In order to use the C code drivers two additional C functions have to be written by the developer, `FlashRead()` is used to tell the software drivers how to read from the memory and `FlashWrite()` is used to tell the software drivers how to write to the memory.

The drivers include example `FlashRead()` and `FlashWrite()` functions and these, combined with setting the correct `BASE_ADDR` macro are suitable for use with the MPC555. The example functions are:

```
#define BASE_ADDR ((volatile unsigned long*)0x00000000L)
.
unsigned long FlashRead( unsigned long ulOff )
{
   return BASE_ADDR[ulOff];
}

static unsigned long FlashWrite( unsigned long ulOff, unsigned long uVal )
{
   return BASE_ADDR[ulOff] = uVal;
}
.
```

The drivers also provide an example of these functions as macros to speed up the code execution (since no function call overhead is required). When used with the MPC555 the `FlashRead()` function can be replaced with the macro:

```
#define FlashRead( ulOff ) ( BASE_ADDR[ulOff] )
```

The `FlashWrite()` function should not be replaced with a macro or an inline function (C++). The use of a function call for each write to the Flash will prevent back-to-back writes from occurring; even with this measure, C programmer should look carefully at the assembled code to ensure that no back-to-back writes occur. The time taken to put the parameters on the stack and make the call will be sufficient for the M58BF008 to recover. In terms of the CPU overhead this causes it will be minimal; programming is the time when the most write accesses to the Flash occur and each double-word takes about 10µs to program compared to an additional 200ns (roughly) in the write access.

CONCLUSION

The M58BF008 can be connected to the MPC555 in a glue-less configuration, providing burst performance. The Flash can be used to boot the MPC555 and does not require shadow RAM, code can be executed directly from the Flash without loss of performance.

REVISION HISTORY

Date	Version	Revision Details
December 2000	-01	First Issue
February 2001	-02	Addition of the DC Characteristics comparison. Other minor corrections

If you have any questions or suggestion concerning the matters raised in this document please send them to the following electronic mail address:

ask.memory@st.com (for general enquiries)

Please remember to include your name, company, location, telephone number and fax number.

littlediode
COMPONENTS FOR ALL

Littlediode supplies new, hard to find or obsolete electronic components and semiconductors all over the world.

With over two million different components listed you are sure to find the part you need.

Feel free to visit us today at our online store:

LittleDiode.com

Looking forward to providing you with the best possible service.